|  |  |
| --- | --- |
|  |  |
| medbase |  |
|  |  |
|  | Giugno 2021ITI “E. Medi” |
|  | Alessio Lustri5° A informatica |

### Prova d’esame a.s. 2020/21

* 1. **Parte 1**

Una struttura ospedaliera polispecialistica vuole registrare l'occupazione dei posti letto nei vari reparti, in modo da monitorare costantemente il numero dei posti disponibili.

Solo il personale dell'Ospedale può accedere al Data Base. con funzioni di accesso personalizzate in base alla mansione ricoperta nella struttura.

* Per ciascun Reparto (Cardiologia, Oncologia, Ostetricia, ...) il direttore generale ha valutato il numero massimo di letti.
* L'ufficio accettazione provvede a ricoverare un Assistito (memorizzato con il solo Nome-Cognome) in un reparto previa verifica della disponibilità.
* Il medico può dimettere un assistito (registrando la data di fine del Ricovero), incrementando così i posti letto del reparto.

Lo studente, fatte le opportune ipotesi aggiuntive, sviluppi

1. Un’analisi della realtà di riferimento individuando la soluzione che a suo motivato giudizio è la più idonea a rispondere alle specifiche indicate;
2. Lo schema concettuale della base di dati;
3. Lo schema logico della base di dati;
4. La descrizione dei vincoli di integrità referenziale e/o vincoli di dominio, laddove presenti;
5. La definizione in linguaggio SQL delle relazioni della base di dati;
6. Le seguenti interrogazioni espresse in linguaggio SQL :
   1. prospetto dei Reparti con dati di occupazione dei letti;
   2. elenco dei reparti senza disponibilità;
   3. tempi medi di permanenza degli assistiti in ciascun reparto.
7. Realizzare un Sito Web che consenta l'interazione con il Data Base da parte dei membri dell'organizzazione e/o degli utenti esterni, utilizzando appropriati linguaggi, sia lato client che lato server. A tale scopo:
   1. Progettare la mappa delle pagine del sito, indicando per ciascuna le azioni svolte, l'eventuale passaggio di parametri e i privilegi richiesti per l'accesso.
   2. Suddividere gli utenti in gruppi, assegnando a ciascuno gli opportuni privilegi.
   3. Descrivere il metodo utilizzato per garantire un accesso controllato al sito.
   4. Costruire un'interfaccia completa per la gestione del sito.

Il sito permette la consultazione dei dati on line, quindi, per il rispetto della privacy, ovvero dei dati degli utenti registrati, per ciascuna query, indicare quali sono i privilegi necessari per l'esecuzione (amministratore, utente, visitatore...).

* 1. **Parte 2**

Il candidato esponga le diverse soluzioni per “ospitare” i contenuti e i servizi implementati nella prima parte, evidenziando i vantaggi e gli svantaggi.

Proponga poi una soluzione adatta al progetto realizzato evidenziando gli aspetti hardware, software e di sicurezza, anche con l’aiuto di diagrammi o schemi.

### Cos’è MedBase

MedBase è il nome dato alla piattaforma per la gestione di strutture sanitaria del committente, questa richiede una serie di istanze fondamentali per il funzionamento:

* Il paziente – colui che viene ricoverato all’interno della struttura,
* Il reparto – dipartimento della struttura dove va ricoverato il paziente,
* Il personale – gli impiegati di ogni tipo della struttura, ognuno potenzialmente con mansioni diverse.

Le entità appena descritte conterranno tutte le informazioni essenziali quali il nome e il cognome dei pazienti, oltre che il codice fiscale. Il reparto avrà oltre che il nome, anche la capienza massima. Il personale invece avrà uno username e una password per effettuare l’accesso alla piattaforma, e un privilegio: ovvero tutte le azioni che un determinato utente può svolgere. Dalla richiesta emerge la necessità di un totale di cinque azioni diverse, ognuna delle quali associata ad un gruppo di utenti:

* Gli analisti, coloro che si occupano di analizzare le statistiche relative a reparti e singoli pazienti.
* I receptionists, il personale addetto all’inserimento dei pazienti non appena arrivano nella struttura, previa verifica di disponibilità di posti letto in quel reparto.
* I medici, il gruppo di utenti che si farà carico dei pazienti e provvederà a dimetterli quando lo riterranno opportuno.
* Il direttore sanitario, la figura che si occupa della gestione dei reparti, aggiungendone di nuovi o rimodulandoli con una diversa capacità di accoglienza.
* Il gestore del personale, la figura incaricata della gestione degli utenti e delle loro possibilità di azione all’interno della piattaforma.

Il sito web della piattaforma prevederà l’accesso direttamente dalla homepage tramite un riquadro a scomparsa. Effettuato l’accesso correttamente, si ritorna alla home e il tasto di login è stato rimpiazzato da un menu a tendina, presente in ogni pagina del sito, da cui si può accedere ad una serie di pagine, ognuna delle quali corrisponde un’azione. Gli utenti che non hanno accesso a determinate funzioni non solo non vedranno comparire i riferimenti a quelle pagine, ma se proveranno ad accedervi in qualunque modo verranno bloccati e visualizzeranno un messaggio di “Accesso non consentito”.

Per ragione di sicurezza, tutte le password verranno criptate secondo l’algoritmo “**sha512**”, che ne impedisce la decodifica e quindi impedisce la criptazione di tutti gli altri dati che necessitano di essere visualizzati così come sono.

MedBase si appoggia ad un database MariaDB, DBMS nato da un fork del più noto MySQL e si interfaccia con l’utente tramite pagine HTML, con l’ausilio dei linguaggi CSS per lo stile grafico dell’applicazione e JavaScript.

Per la creazione di pagine dinamiche e per comunicare con il database è stato utilizzato il linguaggio PHP, mentre per permettere l’invio di mail al team di supporto di Medbase si è utilizzata la libreria PHPmailer.

La piattaforma è indirizzata principalmente ad un target di utenti spesso non propriamente addentro a le dinamiche dei sistemi informatici, si tratta infatti del personale di strutture mediche, per cui la user experience dovrà tenerne conto.

Per permettere la maggior diffusione possibile e per fornire supporto, nella home del sito è stato posto un form di contatto da compilare a cui poi risponderà il team adatto, che si tratti di supporto in caso di problemi che di informazioni in caso qualcuno sia interessato ad implementarla nella propria struttura.

### Analisi delle soluzioni per l’hosting

Per la fruizione pubblica della piattaforma sono state trovate due soluzioni : quella di creare un proprio server da zero, partendo dall’hardware per poi configurare manualmente tutti i software, il cosiddetto “housing”, oppure affidarsi ad una società terza, il provider, che offre servizi simili a pagamento.

|  |  |  |
| --- | --- | --- |
|  | **Vantaggi** | **Svantaggi** |
| **Housing** | * Maggiore versatilità e adattabilità alle proprie esigenze. * Maggior controllo sulle risorse hardware. * Piena autonomia nelle decisioni relative alla sicurezza. | * Richiede competenze in vari ambiti per essere gestito. * Deve essere una macchina totalmente dedicata a quel compito, di conseguenza deve essere accesa h24, cosa che comporta dei costi e porta l’hardware ad un deterioramento prematuro. * Costi di gestione quali corrente e connessione ad Internet. * Richiede di rimanere aggiornati rispetto ai progressi tecnologici. |
| **Hosting** | * Non ci sono costi extra oltre la rata mensile, dato che sono a carico del provider. * Il provider offre anche un servizio clienti che può intervenire in caso di problemi. * Semplice da configurare, spesso tramite interfaccia grafica. | * A meno che non si tratti di un server dedicato di proprietà di terzi, che comporta costi ancor maggiori, solitamente si tratta di un host condiviso: un'unica macchina a disposizione di altre aziende, senza possibilità di sapere con chi condividiamo. * Poco flessibile riguardo le scelte hardware. |

Per Medbase si è optato per un servizio di hosting presso la società tedesca [Hetzner](https://www.hetzner.com/) che offre dei cloud server (server condivisi), completamente configurabili nel software, ne offre con varie caratteristiche in moda da adattarsi alle necessità di ogni persona.

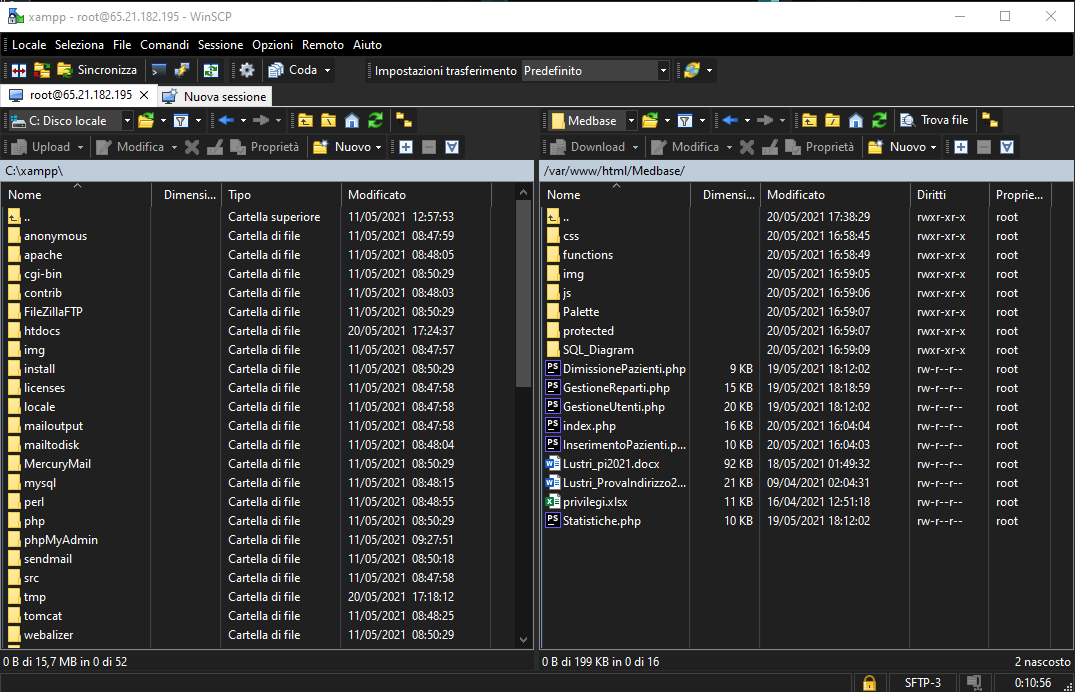
Nel nostro caso il server è dotato del seguente hardware:

* CPU Intel Xeon Platinum 8180
* 2 GB di RAM
* 20 GB di SSD NVMe M.2
* 20 TB di traffico dati

Il server è configurabile anche nella scelta del sistema operativo, nel nostro caso si è optato per Fedora 34, distribuzione GNU/Linux sponsorizzata da Red Hat, particolarmente adatto all’utilizzo dei server per tutta la serie di tool utili a questo scopo.

Ottenuto l’IP del nostro server, che nel nostro caso è situato ad Helsinki, gli si può accedere tramite protocollo SSH, tramite il quale sono stati installati:

* PHP
* phpMyAdmin
* MariaDB
* MariaDB Server
* HTTPD (Apache)

Configurato il database sul server, si è installato il software gratuito WinSCP, un client FTP che tramite interfaccia grafica permette di trasferire i file del sito web sul server.

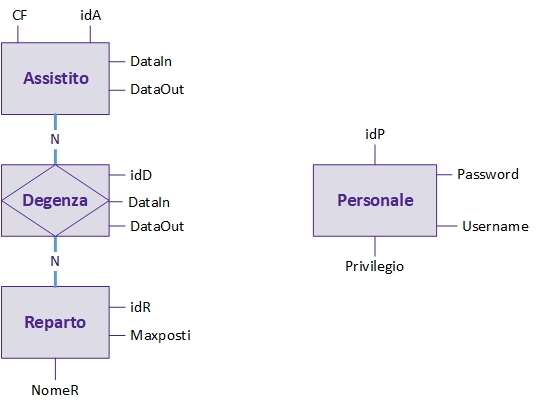
1 - Interfaccia grafica di WinSCP, a sinistra la cartella Xampp in locale, a destra la root del server.

Il dominio del sito web “[alessiolustri.it](https://www.alessiolustri.it)” è stato acquistato dal registrar “[register.it](https://www.register.it)”, il quale fornisce anche una web mail abbinata al dominio.

Per questioni di sicurezza e affidabilità si è deciso di trasferire il controllo del dominio alla società “Cloudflare” che fornisce i certificati SSL, protezione da attacchi DDOS oltre che uno dei DNS server più performanti sul mercato.

Tramite l’interfaccia di Cloudflare è stato definito il sottodominio “[medbase.alessiolustri.it](https://www.medbase.alessiolustri.it)” ed è stato configurato il server Apache per reindirizzare tutti gli utenti che visitano dal sottodominio sopracitato alla sottocartella di Medbase subito sotto la root directory.

### Schema concettuale

Dall’analisi fatta è emerso una associazione di tipo molti a molti tra l’assistito e il reparto in cui viene ricoverato, dato che se un paziente è stato ricoverato e dimesso, può anche tornare nuovamente nella struttura, magari in un nuovo reparto. Per cui, nella tabella nata per scomporre questa associazione, “Degenza”, ad ogni record corrisponderà un ricovero di un paziente in quel ricovero. Se lo stesso paziente viene ricoverato più volte, il suo ID sarà presente in tanti record tanti quanti sono i suoi ricoveri.

Per tutti gli ID non è richiesto un ordine preciso in cui devono essere inseriti, per cui può pensare il DBMS a inserire il loro valore assegnando l’attributo AUTO\_INCREMENT

### Schema logico

*In* ***grassetto le chiavi primarie****, sottolineate le chiavi esterne*.

Reparto (**idR**, nomeR, maxPosti)

Personale (**idP**, username, password, Privilegio)

Assistito (**idA**, nomeA, cognomeA, CF)

Degenza (**idD**, DataIn, DataOut, idA, idR)

### Vincoli

### Vincoli intra-relazionali:

* **Obbligatorietà**: Tutti gli attributi del database sono obbligatori, ad eccezione di DataOut che non può essere riempito finché il paziente non viene dimesso.
* **Vincoli di dominio**:
  + Reparto:
    - NomeR – obbligatorio, massimo 32 caratteri;
    - MaxPosti – obbligatorio, intero;
  + Personale:
    - Username – obbligatorio, massimo 32 caratteri;
    - Password – obbligatorio, lunghezza fissa di 128 caratteri;
    - Privilegio – obbligatorio, intero;
  + Assistito:
    - NomeA - obbligatorio, massimo 32 caratteri;
    - CognomeA - obbligatorio, massimo 32 caratteri;
    - CF – obbligatorio, lunghezza fissa di 16 caratteri;
  + Degenza:
    - DataIn – obbligatorio, data;
    - DataOut – data, di default *null*;
* **Vincoli di tupla**:
  + La DataOut non può essere antecedente a DataIn.
* **Vincoli di unicità**:
  + Un singolo paziente non può essere ricoverato in due reparti contemporaneamente, per cui non ci possono essere due record che fanno riferimento al medesimo assistito e che entrambi abbiano il campo DataOut vuoto.
  + Il campo CF (Codice fiscale) è univoco per ogni paziente.

### Vincoli inter-relazionali:

* **Vincoli di integrità referenziale**:
  + L’ordine in cui le tabelle vengono create deve essere tale in modo che le chiavi esterne siano già esistenti alla creazione della tabella.
  + Il reparto non può più essere eliminato non appena avrà ospitato almeno un paziente, questo perché sarebbe poi impossibile recuperare i dati su quello specifico ricovero per eventuali ricerche statistiche.

### Analisi delle associazioni

* L’associazione tra Assistito e Degenza è di tipo 1-N:
  + Un assistito può avere più degenze nella struttura.
  + Più degenze possono riguardare un singolo paziente.
* L’associazione tra Reparto e Degenza è di tipo 1-N:
  + Un reparto può accogliere più pazienti in diversi ricoveri.
  + Più degenze possono avvenire in un unico reparto.

### Privilegi

|  |  |  |
| --- | --- | --- |
| **Id** | **Mansione** | **Descrizione** |
| 1 | Analista | Visualizzazione e analisi delle statistiche relative a reparti e pazienti. |
| 2 | Receptionist | Inserimento dei pazienti all’interno della piattaforma. |
| 4 | Medico | Dimissione del paziente dai reparti. |
| 8 | Direttore sanitario | Impostazione posti letti massimi e precedenti. |
| 16 | Gestore del personale | Gestione degli account e dei relativi privilegi. |

### Codifica in SQL

### Reparti

**CREATE TABLE IF NOT EXISTS** reparti  
(  
 **idR INT AUTO\_INCREMENT NOT NULL**,  
 **NomeR VARCHAR**(32) **NOT NULL**,  
 **MaxPosti INT NOT NULL**,  
  
 **PRIMARY KEY** (**idR**)  
);  
  
**INSERT INTO** reparti (**NomeR**, **MaxPosti**)

**VALUES** (**'Cardiologia'**, 25);  
**INSERT INTO** reparti (**NomeR**, **MaxPosti**)

**VALUES** (**'Pneumologia'**, 5);

**INSERT INTO** reparti (**NomeR**, **MaxPosti**)

**VALUES** (**'Terapia intensiva'**, 1);

|  |  |  |
| --- | --- | --- |
| **REPARTI** | | |
| **idR** | **NomeR** | **MaxPosti** |
| 1 | Cardiologia | 25 |
| 2 | Pneumologia | 5 |
| 3 | Terapia intensiva | 1 |

### Assistiti

**create table if not exists** assistiti  
(  
 **idA INT AUTO\_INCREMENT NOT NULL**,  
 **nomeA VARCHAR**(32) **NOT NULL**,  
 **cognomeA VARCHAR**(32) **NOT NULL**,  
 **CF CHAR**(16) **NOT NULL**,  
  
 **PRIMARY KEY** (**idA**)  
);  
  
**INSERT INTO** assistiti (**nomeA**, **cognomeA**, **CF**)

**VALUES** (**'Alessio'**, **'Lustri'**, **'LSTLSS02B19F839U'**);  
**INSERT INTO** assistiti (**nomeA**, **cognomeA**, **CF**)

**VALUES** (**'Gabriel'**, **'Amore'**, **'MRAGRL02P09H892Y'**);  
**INSERT INTO** assistiti (**nomeA**, **cognomeA**, **CF**)

**VALUES** (**'Vittorio'**, **'Picone'**, **'PCNVTR02L30F839G'**);

|  |  |  |  |
| --- | --- | --- | --- |
| **ASSISTITI** | | | |
| **idA** | **NomeA** | **CognomeA** | **CF** |
| 1 | Alessio | Lustri | LSTLSS02B19F839U |
| 2 | Gabriel | Amore | MRAGRL02P09H892Y |
| 3 | Vittorio | Picone | PCNVTR02L30F839G |

### Personale

**CREATE TABLE IF NOT EXISTS** personale  
(  
 **idP INT AUTO\_INCREMENT NOT NULL**,  
 **username VARCHAR**(32) **NOT NULL**,  
 **password VARCHAR**(128) **NOT NULL**,  
 **privilegio INT NOT NULL**,  
  
 **PRIMARY KEY** (**idP**)  
);  
  
**INSERT INTO** personale (**username**, **password**, **privilegio**)

**VALUES** (**'admin'**, **'c7ad44cbad762a5da0a452f9e854fdc1e0e7a52a38015f23f3eab1d80b931dd472634dfac71cd34ebc35d16ab7fb8a90c81f975113d6c7538dc69dd8de9077ec'**, 31);  
**INSERT INTO** personale (**username**, **password**, **privilegio**)

**VALUES** (**'analista'**, **'659b83afdc290b9e794af4fb2bec4ebb416ebeecc6efd3f0d957e1f1923e5b088e97bfa41e2385c77d24da9e7f0fd0ca716213052154f7bbc58ecaef45e55843'**, 1);  
**INSERT INTO** personale (**username**, **password**, **privilegio**)

**VALUES** (**'direttore'**, **'b6af56c356c4dc45fea6e06b3703cddca0bd1cba3dc919d71927c37639ec04a61688094334efdd911dad7524cbaeffda2dc7add17459e20a0ab7d7da4880a1f2'**, 8);

|  |  |  |  |
| --- | --- | --- | --- |
| **PERSONALE** | | | |
| **idP** | **username** | **password** | **privilegio** |
| 1 | admin | c7ad44cbad762a5da0a452f9e854fdc1e0  e7a52a38015f23f3eab1d80b931dd4726  34dfac71cd34ebc35d16ab7fb8a90c81f9  75113d6c7538dc69dd8de9077ec | 31 |
| 2 | analista | 659b83afdc290b9e794af4fb2bec4ebb416  ebeecc6efd3f0d957e1f1923e5b088e97bf  a41e2385c77d24da9e7f0fd0ca71621305  2154f7bbc58ecaef45e55843 | 1 |
| 3 | direttore | b6af56c356c4dc45fea6e06b3703cddca0b  d1cba3dc919d71927c37639ec04a616880  94334efdd911dad7524cbaeffda2dc7add1  7459e20a0ab7d7da4880a1f2 | 8 |

### Degenze

**create table if not exists** degenze  
(  
 **idD INT AUTO\_INCREMENT NOT NULL**,  
 **DataIn DATE NOT NULL**,  
 **DataOut DATE**,  
 **idA INT NOT NULL**,  
 **idR INT NOT NULL**,  
  
 **PRIMARY KEY** (**idD**),  
 **FOREIGN KEY** (**idA**) **REFERENCES** assistiti (**idA**),  
 **FOREIGN KEY** (**idR**) **REFERENCES** reparti (**idR**)  
);  
  
**INSERT INTO** degenze (**DataIn**, **DataOut**, **idA**, **idR**)

**VALUES** (**'2020/10/10'**, **'2020/10/20'**, 1, 1);  
**INSERT INTO** degenze (**DataIn**, **DataOut**, **idA**, **idR**)

**VALUES** (**'2020/11/20'**, **'2020/12/10'**, 2, 2);  
**INSERT INTO** degenze (**DataIn**, **DataOut**, **idA**, **idR**)

**VALUES** (**'2021/01/10'**, **'2021/01/15'**, 3, 3);

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PERSONALE** | | | | |
| **idD** | **DataIn** | **DataOut** | **idA** | **idR** |
| 1 | 2020/10/10 | 2020/10/20 | 1 | 1 |
| 2 | 2020/11/20 | 2020/12/10 | 2 | 2 |
| 3 | 2021/01/10 | 2021/01/15 | 3 | 3 |

### Analisi della qualità del database

### Prima forma normale

Il database in oggetto è conforme alla prima forma normale in quanto ogni attributo è definito su un dominio di valori atomici, cioè indivisibili e ogni record ha una propria chiave primaria univoca.

### Seconda forma normale

Il database risulta essere conforme anche seconda forma normale poiché rispetta gli standard del suddetto livello. Infatti, ogni attributo **non** chiave dipende esclusivamente dall’intera chiave primaria del record, e non ad una parte di essa.

### Terza forma normale

Il database è infine conforme anche alla terza forma normale poiché ogni attributo non chiave non dipende in alcun modo da un altro attributo non chiave del record, eliminando così la proprietà della dipendenza transitiva degli attributi.

### Query richieste

### prospetto dei reparti con dati di occupazione dei letti

Per ottenere la tabella richiesta, bisogna prima capire la condizione secondo la quale un ricovero è ancora in corso, oppure è stato terminato. Questa condizione è data dal campo DataOut della tabella Degenze, in caso questo sia *null* significa che il paziente è ancora presente nella struttura.

Applicata questa condizione si va a contare quante sono in quel momento le degenze in corso per ogni reparto, ottenendo il numero di posti letto occupati.

**select** R.NomeR, *count*(D.idD) **as** PostiOccupati, R.MaxPosti,

(MaxPosti-*count*(D.idD)) **as** PostiDisponibili   
**from** degenze D, reparti R  
**where** D.DataOut **is null  
 AND** R.idR = D.idR  
**group by**(D.idR)

|  |  |  |  |
| --- | --- | --- | --- |
| **TABELLA RISULTANTE** | | | |
| **NomeR** | **PostiOccupati** | **MaxPosti** | **PostiDisponibili** |
| Cardiologia | 1 | 25 | 1 |
| Pneumologia | 1 | 5 | 2 |
| Terapia intensiva | 1 | 1 | 3 |

### Elenco dei reparti senza disponibilità;

Similmente alla query precedente, si va a verificare quanti sono i posti letto occupati per ogni reparto e si va ad estrarre il nome del reparto solo dove il numero di posti massimi (MaxPosti) coincide con quello dei posti occupati.

**select** R.NomeR, R.MaxPosti, T1.PostiOccupati  
**from** reparti R,   
 (  
 **select** R.idR, R.NomeR, *count*(D.idD) **as** PostiOccupati  
 **from** degenze D, reparti R  
 **where** D.DataOut **is null  
 AND** R.idR = D.idR  
 **group by**(D.idR)   
 ) **as** T1  
**where** R.MaxPosti = T1.PostiOccupati   
 **and** T1.idR = R.idR

|  |  |  |
| --- | --- | --- |
| **TABELLA RISULTANTE** | | |
| **NomeR** | **MaxPosti** | **PostiOccupati** |
| Terapia intensiva | 1 | 1 |

### tempi medi di permanenza degli assistiti in ciascun reparto.

Per ottenere il prospetto richiesto, è necessario prima di tutto andare ad individuare la durata dei singoli ricoveri con il reparto associato, quando dato verrà poi raggruppato in base al reparto in cui è avvenuto il ricovero e viene calcolata la media con una precisione di due cifre decimali.

**select DISTINCT NomeR**, *truncate*(*AVG*(T1.Durata),2) **as** PermanenzaMedia  
**from** reparti R,  
 (  
 **select idR**, *DATEDIFF*(D.**DataOut**, D.**DataIn**)+1 **as** Durata  
 **from** degenze D  
 **where DataOut is not null** ) **as** T1  
**where** R.**idR** = T1.**idR  
group by** T1.**idR**

|  |  |
| --- | --- |
| **TABELLA RISULTANTE** | |
| **NomeR** | **PermanenzaMedia** |
| Cardiologia | 11.00 |
| Pneumologia | 21.00 |
| Terapia intensiva | 6.00 |

### Sito Web

### Analisi del mercato e del target

#### Scopo della piattaforma:

La piattaforma si propone come uno strumento utile a semplificare la gestione di complesse strutture sanitarie polispecialistiche. La concorrenza, anche nel settore pubblico, è estremamente ricca e questo deve rappresentare un incentivo a sviluppare una piattaforma che abbia il più ampio target possibile, che riceva i feedback migliori possibili dagli utilizzatori ai quali risulti facile l’utilizzo dell’interfaccia.

#### Target:

La piattaforma va sponsorizzata ai dirigenti di strutture sanitarie e a tutti coloro che hanno potere decisionale su quest’ultime. D’altro canto, però la piattaforma va ad essere utilizzata principalmente da medici, che molto spesso non sono addentri alla tecnologia e da personale addetto alla ricezione dei pazienti.

La progettazione dell’interfaccia grafica e quindi del design delle pagine deve essere il più user-friendly possibile.

### Pianificazione e timeline del progetto

Chiarito lo scopo e il target della piattaforma web si è studiata la struttura del sito e si è optato per un sito a maglia intrecciata: questo consente all’utilizzatore di avere, tramite una barra di navigazione posta in alto, un punto di riferimento in qualunque pagina egli si trovi.

Da lì potrà poi raggiungere tutte le pagine a cui ha diritto di accesso in base al suo profilo tramite un menù a tendina oppure modificare la password del proprio account ed eventualmente effettuare il logout, evitando continui cambi di pagina potersi muovere all’interno del sito.

Analizzatele richieste del committente si è convenuto che le azioni del sito si districheranno in un totale di 6 (sei) pagine, con le quali l’utente interagisce.

Le pagine sono le seguenti:

* **Home**, la quale farà da presentazione alla piattaforma e con un modulo di contatto per informazioni o problemi.
* **Statistiche**, tramite la quale gli addetti potranno analizzare informazioni statistiche riguardanti pazienti e reparti.
* **Inserimento paziente**, tramite la quale gli addetti alla ricezione possono inserire i pazienti non appena questi vengono accolti nella struttura.
* **Dimissione paziente**